80

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/838,425	04/19/2001	Frederic Bauchot	FR920000031US1	3575

50170 7590 08/17/2005

IBM CORP. (WIP)
c/o WALDER INTELLECTUAL PROPERTY LAW, P.C.
P.O. BOX 832745
RICHARDSON, TX 75083

EXAMINER	
STEVENS, ROBERT	
ART UNIT	PAPER NUMBER
2176	

DATE MAILED: 08/17/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

47

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 09/838,425
Filing Date: April 19, 2001
Appellant(s): BAUCHOT ET AL.

Stephen J. Walder, Jr.
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed May 24, 2005.

PAGE2

KD

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The brief does not contain a statement identifying the related appeals and interferences which will directly affect or be directly affected by or have a bearing on the decision in the pending appeal is contained in the brief. Therefore, it is presumed that there are none. The Board, however, may exercise its discretion to require an explicit statement as to the existence of any related appeals and interferences.

(3) Status of Claims

The statement of the status of the claims contained in the brief is correct.

(4) Status of Amendments After Final

The appellant's statement of the status of amendments after final rejection contained in the brief is correct.

(5) Summary of Invention

The summary of invention contained in the brief is substantially correct, however the Office provides the following characterization of Appellant's subject matter:

Office Note

Appellant's alleged inventive subject matter describes a mechanism for determining the relationship between a range of cells (i.e., the extent of intersection between two ranges of cell addresses) in a spreadsheet. This relationship is analogous to set theory concepts as visualized using Venn diagrams (which illustrate set relationships such as: intersection, no intersection, "subset of", equality). Appellant alleges that comparison of boolean values (the criteria for setting/unsetting these values and the correlation of these values to a cell/range are unclear), allows one to determine the extent of intersection between cell ranges in a spreadsheet.

(6) Issues

The appellant's statement of the issues in the brief is correct.

(7) Claims Appealed

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Prior Art of Record

Kelly, Julia, Using Microsoft Excel 97, 3rd Edition, Que Corp., Indianapolis, IN, (c) 1998, pp. 124-131, 138-144, 154-189, 209-210 and 337-343.

Deitel, H. M., et al., C++: How to program, 2nd Edition, Prentice Hall, Upper Saddle River, NJ, (c) 1994, pp. 10, 106-110, 147, 243-244, 256-262, 448, 473-479, 483-485, 707-730, 981-987 and 1043-1045.

Microsoft Computer Dictionary, 4th Edition, Microsoft Press, Redmond, WA, (c) 1999, pp. 29, 56-58, 79, 229, 272, 420 and 434.

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 112

The following is a quotation of the first paragraph of 35 U.S.C. 112:

> The specification shall contain a written description of the invention, and of the manner and process of making and using it, in such full, clear, concise, and exact terms as to enable any person skilled in the art to which it pertains, or with which it is most nearly connected, to make and use the same and shall set forth the best mode contemplated by the inventor of carrying out his invention.

Claims 1-7 are rejected under 35 U.S.C. 112, first paragraph, as failing to comply with the enablement requirement. The claim(s) contains subject matter which was not described in the specification in such a way as to enable one skilled in the art to which it pertains, or with which it is most nearly connected, to make and/or use the invention.

Regarding claim 1, the preferred embodiment discloses a technically incorrect solution. According to the procedure disclosed in Fig. 5, the resultant values for Atrue/Afalse/Btrue/Bfalse are ALWAYS True/False/False/True. As such the only possible processing option for step # 515 is #518 "Result=DISJOINTED"

Translated into claim 1 limitations: "determining a first operation ..." is always true (Fig 5, step 506), "determining a second operation ..." is always false (Fig 5, step 511), and "determining whether the first series ..." is always True/False/False/True, always resulting in the DISJOINTED scenario, according to Fig 5 step 515.

Claims 2-5 are rejected by virtue of their dependency upon claim 1.

Claims 6-7 claim the system and computer readable medium subject matter forms which embody claim 1, and therefore are rejected under the same rationale as per claim 1.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 7 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Regarding claim 7, which is dependent upon amended claim 1, the term/phrase "computer readable medium" was not defined in the specification. As such, the scope of this claim is indeterminable.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

Claims 6-7 are rejected under 35 U.S.C. 101 because the claimed invention is directed to non-statutory subject matter.

Regarding independent claim 6: The language of this claim merely describes a computer program per se. As such, this raises a question as to whether the claim is directed merely to an abstract idea that is not tied to a technological art, environment or machine, which would result in a practical application producing a concrete, useful and tangible result to form the basis of statutory subject matter under 35 USC 101.

One technique for satisfying the requirements of 35 USC 101 is to claim code residing in memory (i.e., hardware), wherein that code produces a tangible result.

Regarding claim 7: This claim is directed to a "computer readable medium", which may encompass an intangible embodiment (such as a carrier wave or transmission media).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claims 1-7 are rejected under 35 U.S.C. 103(a) as being unpatentable over Julia Kelly (Using Microsoft Excel 97, 3rd Edition, Que Corp., Indianapolis, IN, © 1998, pp. 138-144 and 154-189, hereafter referred to as "Kelly") in view of H. M. Deitel et al. (C++: How To Program, 2nd Edition, Prentice Hall, Upper Saddle River, NJ, © 1994, pp. 10, 106-110, 147, 243-244, 256-262, 448, 473-479, 483-485, 707-730, 981-987 and 1043-1045, hereafter referred to as "Deitel"). The Microsoft Computer Dictionary, 4th Edition, Microsoft Press, Redmond, WA, © 1999, pp. 29, 56-58, 79, 229, 272, 420 and 434, hereafter referred to as " MS Dictionary", has also been used to supply definitions for various computer terms.

Regarding independent claim 1, Kelly discloses:

> *A method of comparing two series of cells in a multi dimensional spreadsheet comprising a plurality of cells identified by a cell address along each dimension, a series of cells comprising one or a plurality of cell range, a cell range comprising one or a plurality of cells* (pp. 340-343 Figures 18.12-18.14 and especially the Fig. 18.15 comparison summary), *said method comprising the steps of:*

> *defining a boolean attribute, said boolean attribute having a first and a second value;* (pp. 174-175 "IF" section and Fig. 10.23, re: use of nested IF formula. The p. 58 MS Dictionary definition of boolean algebra discloses the well known use of first [i.e., 0] and second [i.e., 1] boolean values as being false and true, respectively.);
>
> *... .*

However, Kelly does not explicitly disclose:

> *...*
>
> *assigning the first value of said boolean attribute to each cell of a first series of cells;*
>
> *assigning the second value of said boolean attribute to each cell of a second series of cells;*
>
> *determining in a first operation whether all the cells of said first series of cells share the same first value of said boolean attribute, or share the same second value of said boolean attribute or do not share a same single value of said boolean attribute;*
>
> *for a second operation, again assigning the first value of said boolean attribute to each cell of the first series of cells;*
>
> *determining in a second operation whether all the cells of the second series of cells share the same first value of said boolean attribute, or share the same second value of said boolean attribute or do not share a same single value of said boolean attribute;*
>
> *recording intermediary information from the first operation and the second operation in a comparison table, stored in a memory of a computer;*
>
> *determining whether the first series and the second series are the same or not by comparing results of the first operation and the second operation:*
>
> *if all the cells of the first series share the same second value of said boolean attribute in said first operation and*
>
> *if all the cells of the second series share the same first value of said boolean attribute in said second operation, the first series and the second series are the same.*

Deitel, though, discloses:

> *...*
>
> *assigning the first value of said boolean attribute to each cell of a first series of cells;* (Assignment is well known in the art. See p. 243 code, esp. line 12.)

> *assigning the second value of said boolean attribute to each cell of a second series of cells;* (Assignment is well known in the art. See p. 243 code, esp. line 12.)
>
> *determining in a first operation whether all the cells of said first series of cells share the same first value of said boolean attribute, or share the same second value of said boolean attribute or do not share a same single value of said boolean attribute;* (p. 476 code block lines 102-114, esp. line 113 which is executed only when the compared array cells [i.e., series] are equal [i.e., the same])
>
> *for a second operation, again assigning the first value of said boolean attribute to each cell of the first series of cells;* (Assignment is well known in the art. See p. 243 code, esp. line 12.)
>
> *determining in a second operation whether all the cells of the second series of cells share the same first value of said boolean attribute, or share the same second value of said boolean attribute or do not share a same single value of said boolean attribute;* (p. 476 code block lines 102-114, esp. line 113 which is executed only when the compared array cells [i.e., series] are equal [i.e., the same])
>
> *recording intermediary information from the first operation and the second operation in a comparison table, stored in a memory of a computer;* (It is a matter of obvious design choice to create an additional array/table to store results. Pp. 243-244 code lines 12, 23 and 41-45 disclose creating and modifying [i.e., recording results in] an array data structure.)
>
> *determining whether the first series and the second series are the same or not by comparing results of the first operation and the second operation:* (p. 476 code block lines 102-114, esp. line 113 which is executed only when the compared array cells [i.e., series] are equal [i.e., the same])
>
> *if all the cells of the first series share the same second value of said boolean attribute in said first operation and if all the cells of the second series share the same first value of said boolean attribute in said second operation, the first series and the second series are the same.* (p. 476 code block 109-113 discloses comparing array cells for equality/similarity)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Deitel for the benefit of Kelly, because to do so would enable a programmer to build reusable software components, as taught by Deitel in the

2nd paragraph on p. 10 after “Portability Tip 1.1”. These references were all applicable to the same field of endeavor, i.e., application program use and development.

Regarding claim 2, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

However, Kelly does not explicitly disclose:

> *wherein the step of determining whether the first series and the second series are the same or not comprises the further step of determining whether the first series and the second series are disjoined or not by comparing the results of the first operation and the second operation:*
>
> > *if all the cells of the first series share the same first value of said boolean attribute in said first operation and if all the cells of the second series share the same second value of said boolean attribute in said second operation, the first series and the second series are disjoined.*

Deitel, though, discloses:

> *wherein the step of determining whether the first series and the second series are the same or not comprises the further step of determining whether the first series and the second series are disjoined or not by comparing the results of the first operation and the second operation:*
>
> > *if all the cells of the first series share the same first value of said boolean attribute in said first operation and if all the cells of the second series share the same second value of said boolean attribute in said second operation, the first series and the second series are disjoined.* (p. 476 code block on lines 102-114, esp. line 111, which returns when the compared ranges are not equal [i.e., disjoined])

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Deitel for the benefit of Kelly, because to do so would enable a programmer to build reusable software components, as taught by Deitel in the

2nd paragraph on p. 10 after "Portability Tip 1.1". These references were all applicable to the same field of endeavor, i.e., application program use and development.

Regarding claim 3, which is dependent upon claim 2, the limitations of claim 2 have been previously addressed.

However, Kelly does not explicitly disclose:

> *wherein the step of determining whether the first series and the second series are the same or not, comprises the further step of determining whether the first series and the second series overlap or not by comparing the results of the first operation and the second operation;*
>
> *if all the cells of the first series do not share the same single value of said boolean attribute in said first operation and if all the cells of the second series do not share the same single value of said boolean attribute in said second operation, the first series and the second series overlap.*

Deitel, though, discloses:

> *wherein the step of determining whether the first series and the second series are the same or not, comprises the further step of determining whether the first series and the second series overlap or not by comparing the results of the first operation and the second operation;*
>
> *if all the cells of the first series do not share the same single value of said boolean attribute in said first operation and if all the cells of the second series do not share the same single value of said boolean attribute in said second operation, the first series and the second series overlap.* (pp. 985-986 Fig. 20.31 code, esp. lines 21-38 disclosing comparison of values to table entries to determine if those values are found/included in the table [i.e., determine if there is overlap or a subset relationship). It is a matter of obvious design choice that the "values" can come from another table.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Deitel for the benefit of Kelly, because to do so would enable a programmer to build reusable software components, as taught by Deitel in the

2nd paragraph on p. 10 after "Portability Tip 1.1". These references were all applicable to the same field of endeavor, i.e., application program use and development.

Regarding claim 4, which is dependent upon claim 3, the limitations of claim 3 have been previously addressed.

However, Kelly does not explicitly disclose:

> *wherein the step of determining whether the first series and the second series are the same or not, comprises the further step of determining whether the first series and the second series are included one in the other or not by comparing the results of the first operation and the second operation:*
>
> > *if all the cells of the first series share the same second value of said boolean attribute in said first operation and if all the cells of the second series do not share the same single value of said boolean attribute in said second operation, the first series is included in the second series;*
> >
> > *if all the cells of the first series do not share the same single value of said boolean attribute in said first operation and if all the cells of the second series share the same first value of said boolean attribute in said second operation, the second series is included in the first series.*

Deitel, though, discloses:

> *wherein the step of determining whether the first series and the second series are the same or not, comprises the further step of determining whether the first series and the second series are included one in the other or not by comparing the results of the first operation and the second operation:*
>
> > *if all the cells of the first series share the same second value of said boolean attribute in said first operation and if all the cells of the second series do not share the same single value of said boolean attribute in said second operation, the first series is included in the second series;*
> >
> > *if all the cells of the first series do not share the same single value of said boolean attribute in said first operation and if all the cells of the second series share the same first value of said boolean attribute in said second operation, the second series is included in the first series.* (pp. 985-986 Fig. 20.31 code, esp. lines 21-38 disclosing comparison of values to table entries to determine if those values are found/included in the table

> [i.e., determine if there is overlap or a subset relationship). It is a matter of obvious design choice that the "values" can come from another table.)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Deitel for the benefit of Kelly, because to do so would enable a programmer to build reusable software components, as taught by Deitel in the 2nd paragraph on p. 10 after "Portability Tip 1.1". These references were all applicable to the same field of endeavor, i.e., application program use and development.

Regarding claim 5, which is dependent upon claim 1, the limitations of claim 1 have been previously addressed.

However, Kelly does not explicitly disclose:

> *wherein said boolean attribute is temporary.*

Deitel, though, discloses:

> *wherein said boolean attribute is temporary.* (p. 147 Fig. 20.31 "3.4 Functions" 1st sentence discussing local variables. It is well known in the art that the scope of local variables is limited [i.e., temporary].)

It would have been obvious to one of ordinary skill in the art at the time of the invention to apply the teachings of Deitel for the benefit of Kelly, because to do so would enable a programmer to build reusable software components, as taught by Deitel in the 2nd paragraph on p. 10 after "Portability Tip 1.1". These references were all applicable to the same field of endeavor, i.e., application program use and development.

Claim 6 is directed to a system comprising the means for implementing the method set forth in claim 1. As such, claim 6 is substantially similar to claim 1, and therefore likewise rejected.

Claim 7 is directed to a computer readable medium comprising instructions for implementing the method set forth in claim 1. As such, claim 7 is substantially similar to claim 1, and therefore likewise rejected.

(10) Response to Argument

Beginning on p. 7 of the Appeal Brief (hereinafter "the Brief"), Appellant presents the following issues, which are accordingly addressed below.

a. "Rejections of Claim 11 under 35 USC §112, First Paragraph" (pages 7-10, argument A of the Brief)

Appellant argues that the Office misinterprets Fig. 5, as filed. Appellant asserts that this figure sets forth an enabling disclosure of the inventive subject matter.

The Office respectfully maintains the argument that Fig. 5 is defective on its face. The purpose of the alleged inventive subject matter was purportedly to indicate one of five conditions (OVERLAP, B_IN_A, DISJOINED, SAME, or A_IN_B) regarding cell range intersection. However, only one such condition (DISJOINED) will always be indicated, based on the logic presented in Fig. 5. Therefore, the Appellant has not only failed to enable Appellant's asserted invention, but has apparently described a non-working embodiment, since a comparison of any two cell ranges (regardless of their extent of overlap) will always appear to be DISJOINED, based upon Appellant's disclosure.

A walk through the flow chart of Fig. 5 clearly shows that one skilled in the art ALWAYS arrives at the same result for the disclosed Flow chart. Referring to Fig. 5:

step 504: initialize variables Atrue, Afalse, Btrue and Bfalse to F/F/F/F .

step 505: initialize Aserie.TempAttribute to True and Bserie.TempAttribute to False .

step 506: check the value of Aserie.TempAttribute. (It was just set in step 505 to True, therefore step 507 is always executed next. Note that step 509 is never executed. Step 508 is also never executed, and does not make sense. How could this variable have an undetermined value, when its value was explicitly set in step 505?)

step 507: set the variable Atrue to T, the result being that the variables Atrue, Afalse, Btrue and Bfalse are now equal to T/F/F/F .

step 510: re- initialize Aserie.TempAttribute to True . (This does not make sense. This variable was previously set to "True" in step 505, and its value has not been changed prior to step 510.)

step 511: check the value of Bserie.TempAttribute. (It was previously set in step 505 to False, therefore step 514 is always executed next. Note that step 512 is never executed. Step 513 is also never executed, and does not make sense. How could this variable have an undetermined value, when its value was explicitly set in step 505?)

step 514: set the variable Bfalse to T, the result being that the variables Atrue, Afalse, Btrue and Bfalse are now equal to T/F/F/T .

step 515: check the values of the variables Atrue, Afalse, Btrue and Bfalse. As indicated in step 514 above, these values are: T/F/F/T .

steps 516, 517, 519 and 520: These steps will NEVER be executed.

step 518: "Result" is ALWAYS set to DISJOINED, because Atrue, Afalse, Btrue and Bfalse will ALWAYS equal T/F/F/T, based upon Appellant's disclosed Fig. 5.

The Office notes that there is no other possible path through the Fig. 5 flow chart. Appellant has not enabled Appellant's purported inventive subject matter. The Office therefore maintains the rejections of claims 1-7 under 35 USC §112, First Paragraph.

b. "Rejections of Claim 7 under 35 USC §112, Second Paragraph" (pages 10-12, argument B of the Brief)

Appellant first asserts that the terminology "computer usable medium" is well known in the art, and therefore the rejection of claim 11 under 35 USC §112, 2nd paragraph is inappropriate. Appellant further provides examples of Appellant's interpretation of this terminology to include, inter alia, "floppy disks, hard disks, magnetic tape, CD-ROMs, DVD-ROMs, carrier waves, transmission media and the like".

The Office first notes that the actual claim language is directed to a "computer readable medium". Therefore, the Office's response is directed to a computer readable medium (vice computer usable medium).

The Office also notes that no evidence exists within the "as-filed" specification to support Appellant's definition of computer readable medium (i.e., as encompassing transmission media). In fact, the terminology does not appear within the "as-filed" specification at all. One could fairly interpret computer readable media as encompassing a piece of paper (having code embedded thereon and for scanning into a computer). Additionally, Appellant's interpretation of this undefined terminology creates a further question as to whether the claim is statutory under 35 USC §101, as addressed below. This terminology is therefore vague and ambiguous, and thus renders the claim indefinite.

It would unfair to the public to permit one to purposefully not define specification terminology, and then subsequently expand the scope of the undefined terminology at a later date. Such practice does not put the public on notice as to the metes and bounds of the claimed subject matter.

The Office therefore maintains the rejection of claim 7 under 35 USC §112, Second paragraph.

c. "Rejection of Claims 6-7 under 35 USC §101" (pages 12-14, argument C of the Brief)

Regarding claim 6, Appellant merely asserts that using the terminology "system" and "means for" renders a claim tangibly embodied, and thus statutory under 35 USC §101.

The Office respectfully notes, however, that the recited language of claim 6 requires no hardware. A "system" may encompass a purely software system. "Means for" may encompass one or more software routines. Because no hardware is required, this claim is not tangibly embodied (i.e., encompasses at least one intangible embodiment) and thus not statutory under 35 USC §101.

Regarding claim 7, Appellant argues for the classification of carrier waves and transmission media as being tangible media and thus statutory under 35 USC §101. As support, Appellant asserts (at the bottom of page 12 and carrying over to page 13) an MPEP passage stating "When functional descriptive material is <u>recorded</u> on some computer-readable medium, it becomes structurally and functionally interrelated to the medium and will be statutory in most cases since use of technology permits the function of the descriptive material to be realized." See MPEP 2106 (IV)(B)(1).

The Office first notes that Appellant's cited passage requires a "recording". Carrier waves are transitory phenomena and thus do not record the functional descriptive material discussed in this passage. Secondly, the cited passage also clearly indicates that even claiming functional descriptive material on a computer-readable medium does not guarantee that the claim will be statutory.

Appellant has repeatedly stated (via amendment, after final, and appeal brief submissions) that Appellant's intention is to claim an intangible embodiment, such as a carrier wave or transmission medium. A signal is nonstatutory even if it has a physical existence, because it is not tangible physical matter. Non-tangible physical phenomena are not the types of subject matter which were intended to fall within the scope of patentable subject matter under 35 USC §101.

Under current policy, the Office requires a tangible embodiment for a claim to be statutory under 35 USC §101. For at least the rationale provided above, the rejection of claim 7 under 35 USC §101 is proper.

The Office therefore maintains the rejections of claims 6-7 under 35 USC §101.

d. "Rejection of Claims 1-7 under 35 USC §103(a)" (pages 14-23, argument D of the Brief)

Pages 14-21: Regarding Claims 1, 6 and 7, Appellant asserts that neither Kelly nor Deitel teach the claimed limitations.

The Office respectfully notes that the recited limitations are merely a laundry list of operations that are exceedingly well known to those skilled in the art. The claims recite multiple assignment steps, multiple comparison ("determining ... ") steps, and a storing ("recording ...") step. All of these concepts are exceedingly well known. Performing such operations on a specifically named variable, using a specific data type, or employing data structures having specific formats, for instance, is merely a matter of obvious design choice. In other words, it would have been obvious to one skilled in the art to change names (e.g., series vice range), or data types (e.g., boolean vice integer) or data structures (group of cells vice array) used in the design of any system.

Furthermore, the IF-THEN-OTHERWISE function of Kelly illustrates a boolean implementation embedded within an IF statement construct. In order to make a determination as to "whether a condition is met", a boolean variable value must have been established (i.e., set, unset or "managed", to use Appellant's lingo). This is inherent/implicit in computer operations. A variable, represented by an address space in computer memory, has been assigned a value and that value is evaluated for a certain condition as per the teachings of Kelly. This is implicitly/inherently (i.e., as an underlying computer science concept) taught by Kelly.

Additionally, Appellant appears to be arguing that neither reference anticipates claim 1. The Office, though, notes that the rejection is based upon the obviousness not anticipation. Obviousness, and, for that matter, anticipation (e.g., regarding inherency), does not mandate that the cited passage in the prior art mimic the exact claim language. Refer back to the first paragraph in this section discussing obviousness, design choice and system design.

As for the ability to combine the teachings of the cited references, both Kelly and Deitel have been cited for their teachings of well known computer concepts. There is no requirement for the Office to build Appellant's purported subject matter.

The Office notes that claims 6 and 7 are substantially similar to claim 1, and therefore likewise rejected. The Office thus maintains the rejections of claims 1, 6 and 7 under 35 USC 35 USC §103(a).

Page 21, Regarding claims 2-5, Appellant alleges that these claims are patentable by virtue of their dependency on claim 1.

The Office respectfully disagrees with Appellant's assertion for the reasons presented above.

Page 21: Regarding claim 2, Appellant alleges that Kelly and Deitel do not teach the comparison of boolean values.

The Office notes that the comparison of any pair of values is exceedingly well known in the art. The data type of the compared values is merely a matter of obvious design choice. The number of these values contained in a particular data structure, (such as an array, list or even a single variable, for that matter) is also merely a matter of obvious design choice.

As previously stated, performing such operations on a specifically named variable, using a specific data type, or employing data structures having specific formats, for instance, is merely a matter of obvious design choice. In other words, it would have been obvious to one skilled in the art to change names (e.g., series vice range), or data types (e.g., boolean vice integer) or data structures (group of cells vice array) used in the design of any system.

Furthermore, performing an action based upon the results of a comparison is also exceedingly well known in the art. By way of example, the IF-THEN-OTHERWISE function disclosed on pages 174-175 of Kelly clearly illustrates an action being taken upon the determination of the result of a comparison operation. (I.e., a letter grade is assigned based upon comparison of a test score to a grade scale.) The Office notes that the specific values compared and the resulting action taken are merely matters of obvious design choice.

The Office therefore maintains the rejection of claim 2 under 35 USC 35 USC §103(a).

Pages 21-22: Regarding claim 3, Appellant alleges that Kelly and Deitel do not teach the comparison of boolean values.

The Office notes that the comparison of any pair of values is exceedingly well known in the art. The data type of the compared values is merely a matter of obvious design choice. The number of these values contained in a particular data structure, (such as an array, list or even a single variable, for that matter) is also merely a matter of obvious design choice.

As previously stated, performing such operations on a specifically named variable, using a specific data type, or employing data structures having specific formats, for instance, is merely a matter of obvious design choice. In other words, it would have been obvious to one skilled in the art to change names (e.g., series vice range), or data types (e.g., boolean vice integer) or data structures (group of cells vice array) used in the design of any system.

Furthermore, performing an action based upon the results of a comparison is also exceedingly well known in the art. By way of example, the IF-THEN-OTHERWISE function disclosed on pages 174-175 of Kelly clearly illustrates an action being taken upon the determination of the result of a comparison operation. (I.e., a letter grade is assigned based upon comparison of a test score to a grade scale.) The Office notes that the specific values compared and the resulting action taken are merely matters of obvious design choice.

The Office therefore maintains the rejection of claim 3 under 35 USC 35 USC §103(a).

<u>Page 22: Regarding claim 4, Appellant alleges that Kelly and Deitel do not teach the comparison of boolean values.</u>

The Office notes that the comparison of any pair of values is exceedingly well known in the art. The data type of the compared values is merely a matter of obvious design choice. The number of these values contained in a particular data structure, (such as an array, list or even a single variable, for that matter) is also merely a matter of obvious design choice.

As previously stated, performing such operations on a specifically named variable, using a specific data type, or employing data structures having specific formats, for instance, is merely a matter of obvious design choice. In other words, it would have been obvious to one skilled in the art to change names (e.g., series vice range), or data types (e.g., boolean vice integer) or data structures (group of cells vice array) used in the design of any system.

Furthermore, performing an action based upon the results of a comparison is also exceedingly well known in the art. By way of example, the IF-THEN-OTHERWISE function disclosed on pages 174-175 of Kelly clearly illustrates an action being taken upon the determination of the result of a comparison operation. (I.e., a letter grade is assigned based upon comparison of a test score to a grade scale.) The Office notes that the specific values compared and the resulting action taken are merely matters of obvious design choice.

The Office therefore maintains the rejection of claim 4 under 35 USC 35 USC §103(a).

Pages 22-23: Regarding claim 5, Appellant alleges that Kelly and Deitel do not teach the use of temporary boolean values.

The Office respectfully notes that the addition of the term "temporary" adds nothing to the recited limitation. It's unclear what Appellant is thinking here. Appellant appears to be engaging in a form over substance exercise. An attribute, as used by Appellant, is merely a variable. All variable values are temporary. That's the meaning of the word "variable" – it can be changed. Therefore the

Furthermore, Appellant appears to be selectively redacting the Office's prior art citations. As explained in the citation, it is exceedingly well known by one skilled in the art that the scope of local variables is temporary. Deitel has once again been cited to show this well known computer programming concept of temporary values (aka temporary variables, temporary attributes, etc.).

The Office therefore maintains the rejection of claim 5 under 35 USC 35 USC §103(a).

For at least the reasons discussed above, the Office maintains the rejections of claims 1-7 under 35 USC §103(a).

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Robert Stevens

Robert Stevens
August 3, 2005

William L. Bashore
WILLIAM BASHORE
PRIMARY EXAMINER
8/3/2005

Conferees

Heather Herndon

Stephen Hong

IBM CORP. (WIP)
c/o WALDER INTELLECTUAL PROPERTY LAW, P.C.
P.O. BOX 832745
RICHARDSON, TX 75083